

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 3 月 1 3 日
Date of Application:

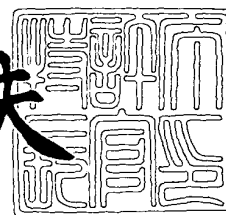
出 願 番 号 特 願 2 0 0 3 - 0 6 8 5 5 3
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 0 6 8 5 5 3]

出 願 人 株式会社日立製作所
Applicant(s):

2 0 0 3 年 7 月 3 0 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 KN1519

【提出日】 平成15年 3月13日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/00

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社
日立製作所 システム開発研究所内

【氏名】 中谷 洋司

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社
日立製作所 システム開発研究所内

【氏名】 岩寄 正明

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社
日立製作所 システム開発研究所内

【氏名】 円光 豊

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100093492

【弁理士】

【氏名又は名称】 鈴木 市郎

【電話番号】 03-3591-8550

【選任した代理人】

【識別番号】 100078134

【弁理士】

【氏名又は名称】 武 顕次郎

【手数料の表示】**【予納台帳番号】** 113584**【納付金額】** 21,000円**【提出物件の目録】****【物件名】** 明細書 1**【物件名】** 図面 1**【物件名】** 要約書 1**【プルーフの要否】** 要

【書類名】 明細書

【発明の名称】 分散ファイルシステム、分散ファイルシステムサーバ及び分散ファイルシステムへのアクセス方法

【特許請求の範囲】

【請求項 1】 集中型のファイルシステムにアクセスを行うためのプロトコルを用いてアクセス可能な分散ファイルシステムにおいて、分散してファイルを格納している複数の D F S サーバをネットワーク上に備え、前記複数の D F S サーバの少なくとも 1 つは、集中型のファイルシステムにアクセスを行うためのプロトコルを分散ファイルをアクセス可能なプロトコルに変換して分散ファイルにアクセスするゲートウェイ部を備えて構成されることを特徴とする分散ファイルシステム。

【請求項 2】 前記ゲートウェイ部は、集中型のファイルシステムに対する要求から、分散ファイルシステムに対する要求を作成して分散ファイルシステムにアクセスを行い、分散ファイルシステムに対する要求の結果から集中型のネットワークファイルシステムに対する要求の結果を作成することを特徴とする請求項 1 記載の分散ファイルシステム。

【請求項 3】 前記ゲートウェイ部は、ディレクトリ構造を持ち、ファイルの存在するディレクトリ位置とディレクトリ内で一意のファイル名とによってファイルを特定するファイルシステムに対するアクセスを行うための要求を受け付け、ディレクトリ構造を持たず、位置に関係なくシステム内で一意につけられた識別子を用いてファイルを特定する分散ファイルシステムにアクセスを行うことを特徴とする請求項 1 記載の分散ファイルシステム。

【請求項 4】 前記ゲートウェイ部は、ディレクトリ操作をエミュレートし、ファイルのディレクトリ位置とファイル名とからファイルシステム上でファイルを一意に識別するためのファイルの識別子を求めることを特徴とする請求項 3 記載の分散ファイルシステム。

【請求項 5】 前記ゲートウェイ部は、1 度書き込んだファイルに対して内容を更新することが可能であるファイルシステムに対してアクセスを行うための要求を受け付け、1 度書き込んだファイルに対して内容の更新を行うことが不可

能で、ファイルの内容の変更を、ファイルの新しい世代を作成し、作成された新しい世代に対して書き込むことにより行うことを特徴とする請求項1記載の分散ファイルシステム。

【請求項6】 前記ゲートウェイ部は、ファイルに対する参照要求に対し、ファイルシステムから最新の世代を求めてアクセスを行うことを特徴とする請求項5記載の分散ファイルシステム。

【請求項7】 前記ゲートウェイ部は、複数の要求からなる一連のファイルの内容の変更要求を受け付け、1つの世代に変更内容を書き込み、一連の変更要求の終了を認識して世代の作成を完了することを特徴とする請求項5記載の分散ファイルシステム。

【請求項8】 集中型のファイルシステムにアクセスを行うためのプロトコルを用いてアクセス可能な分散ファイルシステムの分散してファイルを格納している分散ファイルシステムサーバにおいて、集中型のファイルシステムにアクセスを行うためのプロトコルを分散ファイルをアクセス可能なプロトコルに変換して分散ファイルにアクセスするゲートウェイ部を備えて構成されることを特徴とする分散ファイルシステムサーバ。

【請求項9】 前記ゲートウェイ部は、集中型のファイルシステムに対する要求から、分散ファイルシステムに対する要求を作成して分散ファイルシステムにアクセスを行い、分散ファイルシステムに対する要求の結果から集中型のネットワークファイルシステムに対する要求の結果を作成することを特徴とする請求項8記載の分散ファイルシステムサーバ。

【請求項10】 前記ゲートウェイ部は、ディレクトリ構造を持ち、ファイルの存在するディレクトリ位置とディレクトリ内で一意のファイル名とによってファイルを特定するファイルシステムに対するアクセスを行うための要求を受け付け、ディレクトリ構造を持たず、位置に関係なくシステム内で一意につけられた識別子を用いてファイルを特定する分散ファイルシステムにアクセスを行うことを特徴とする請求項8記載の分散ファイルシステムサーバ。

【請求項11】 前記ゲートウェイ部は、ディレクトリ操作をエミュレートし、ファイルのディレクトリ位置とファイル名とからファイルシステム上でファ

イルを一意に識別するためのファイルの識別子を求めることを特徴とする請求項 1 0 記載の分散ファイルシステムサーバ。

【請求項 1 2】 前記ゲートウェイ部は、1 度書き込んだファイルに対して内容を更新することが可能であるファイルシステムに対してアクセスを行うための要求を受け付け、1 度書き込んだファイルに対して内容の更新を行うことが不可能で、ファイルの内容の変更を、ファイルの新しい世代を作成し、作成された新しい世代に対して書き込むことにより行うことを特徴とする請求項 8 記載の分散ファイルシステムサーバ。

【請求項 1 3】 前記ゲートウェイ部は、ファイルに対する参照要求に対し、ファイルシステムから最新の世代を求めてアクセスを行うことを特徴とする請求項 1 2 記載の分散ファイルシステムサーバ。

【請求項 1 4】 前記ゲートウェイ部は、複数の要求からなる一連のファイルの内容の変更要求を受け付け、1 つの世代に変更内容を書き込み、一連の変更要求の終了を認識して世代の作成を完了することを特徴とする請求項 1 2 記載の分散ファイルシステムサーバ。

【請求項 1 5】 集中型のファイルシステムにアクセスを行うためのプロトコルを用いて、分散ファイルシステムにアクセスを行う分散ファイルシステムへのアクセス方法において、集中型のファイルシステムにアクセスを行うためのプロトコルを分散ファイルをアクセス可能なプロトコルに変換して分散ファイルにアクセスすることを特徴とする分散ファイルシステムへのアクセス方法。

【請求項 1 6】 前記プロトコルの変換が、集中型のファイルシステムに対する要求から、分散ファイルシステムに対する要求を作成し、分散ファイルシステムに対する要求の結果から集中型のネットワークファイルシステムに対する要求の結果を作成する変換であることを特徴とする請求項 1 5 記載の分散ファイルシステムへのアクセス方法。

【請求項 1 7】 前記プロトコル変換が、ディレクトリ構造を持ち、ファイルの存在するディレクトリ位置とディレクトリ内で一意のファイル名とによってファイルを特定するファイルシステムに対するアクセスを行うための要求を、ディレクトリ構造を持たず、位置に関係なくシステム内で一意につけられた識別子

を用いたアクセス要求への変換であることを特徴とする請求項 15 記載の分散ファイルシステムへのアクセス方法。

【請求項 18】 前記要求の変換が、ディレクトリ操作をエミュレートし、ファイルのディレクトリ位置とファイル名とからファイルシステム上でファイルを一意に識別するためのファイルの識別子を求めることにより行われることを特徴とする請求項 17 記載の分散ファイルシステムへのアクセス方法。

【請求項 19】 前プロトコル変換が、1 度書き込んだファイルに対して内容を更新することが可能であるファイルシステムに対してアクセスを行うための要求を、1 度書き込んだファイルに対して内容の更新を行うことが不可能で、ファイルの内容の変更を、ファイルの新しい世代を作成し、作成された新しい世代に対して書き込む要求への変換であることを特徴とする請求項 15 記載の分散ファイルシステムへのアクセス方法。

【請求項 20】 前記要求の変換が、ファイルに対する参照要求に対し、ファイルシステムから最新の世代を求めてアクセスを行う要求への変換であることを特徴とする請求項 19 記載の分散ファイルシステムへのアクセス方法。

【請求項 21】 集中型のファイルシステムにアクセスを行うためのプロトコルを用いて、分散ファイルシステムにアクセスを行う分散ファイルシステムへのアクセスを行う処理プログラムであって、ファイルの参照時、ディレクトリ構造を持ち、ファイルの存在するディレクトリ位置とディレクトリ内で一意のファイル名とによってファイルを特定する情報を、ディレクトリ構造を持たず、位置に関係なくシステム内で一意につけられた識別子に変換するステップと、前記識別子を用いてファイルをアクセスするステップと、参照のために読み出したデータを応答として返すステップとを有することを特徴とする処理プログラム。

【請求項 22】 集中型のファイルシステムにアクセスを行うためのプロトコルを用いて、分散ファイルシステムにアクセスを行う分散ファイルシステムへのアクセスを行う処理プログラムであって、ファイルの更新時、すでに更新中のデータの更新であるか否かを判定するステップと、すでに更新中のデータの更新ではない場合に、更新ファイルの識別子を得るステップと、ファイルデータ更新中を登録するステップと、前記識別子を用いてファイルをアクセスしてデータ書

き込みを要求するステップとを有することを特徴とする処理プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、分散ファイルシステム（DFS：ディストリビューティッドファイルシステム）、分散ファイルシステムサーバ及び分散ファイルシステムへのアクセス方法に係り、特に、ネットワーク上の複数のサーバにファイルを分散して格納することにより、1つのファイルシステムを構成する分散ファイルシステム、分散ファイルシステムサーバ及び分散ファイルシステムへのアクセス方法に関する。

【0002】

【従来の技術】

ネットワークファイルシステムとして、NFSやCIFS（コモンインターネットファイルシステム）と呼ばれるUNIX（登録商標）等のOS上に構築されたファイルシステムが知られている。これらのファイルシステムは、1つのサーバによって1つのファイルシステムを構成する集中型のファイルシステムであり、ファイルの実体がある特定のサーバ上に存在する。クライアントからファイルに対してアクセスを行う場合、各クライアントは、それぞれのプロトコルを用いてまずこのファイルの存在するサーバにアクセスを行う。そして、クライアントは、サーバ上でファイルを特定するために、ディレクトリ構造を用いる。

【0003】

これに対して、OceanStore等のDFSは、クライアントがファイルにアクセスを行う場合、サーバやパス名を特定する代わりに、GUIDというシステムで一意に付けられた識別子を用いている。DFSの場合、ファイルは、ネットワーク上に存在する複数のDFSサーバ上のいずれかにその実体が存在する。ファイルの実体は、1つのサーバだけが持っている必要はなく、複製として他のサーバ上に存在してもよい。また、ファイルの実体は、そのファイル全体を1つのDFSサーバで保持する必要はなく、ファイルが幾つかの部分に分割され、1つのDFSサーバにはその分割された1つの部分（フラグメントと呼ぶ）のみを存在させ、他

のフラグメントを他のDFSサーバ上に存在させてもよい。

【0004】

DFSにおけるファイルにアクセスするクライアントは、ファイルに対する参照、書き込みを行う場合、ネットワーク上のいずれか1つのサーバに対し、GUIDを指定してファイルを特定してアクセスする。

【0005】

なお、本発明に近い従来技術として、ストレージサーバがあらゆる種類のユーザデータへの接続を担う通信インタフェースを備えたシステムが、例えば、特許文献1等に記載されて知られている。

【0006】

【特許文献1】

特開2000-339098号公報

【0007】

【発明が解決しようとする課題】

DFSは、NFSやCIFSといったネットワークファイルシステムと異なる特性を持つため、DFSのファイルにアクセスを行うためには、専用のDFSプロトコルを用いてアクセスを行わなければならない。このため、NFSやCIFSといったプロトコルのみを使用する従来のクライアントは、DFSのファイルに対してアクセスを行うことが不可能であり、従来から用いていたプログラムを改良する等、クライアント側をDFSに対応させる変更を行う必要があった。

【0008】

本発明の目的は、前述した点に鑑み、クライアント側の変更を行うことなく、従来プロトコルを用いてDFSのファイルにアクセスを行うことを可能とした分散ファイルシステム、分散ファイルシステムサーバ及び分散ファイルシステムへのアクセス方法を提供することにある。

【0009】

【課題を解決するための手段】

本発明によれば前記目的は、集中型のファイルシステムにアクセスを行うためのプロトコルを用いてアクセス可能な分散ファイルシステムにおいて、分散して

ファイルを格納している複数のDFSサーバをネットワーク上に備え、前記複数のDFSサーバの少なくとも1つが、集中型のファイルシステムにアクセスを行うためのプロトコルを分散ファイルをアクセス可能なプロトコルに変換して分散ファイルにアクセスするゲートウェイ部を備えて構成されることにより達成される。

【0010】

また、前記目的は、集中型のファイルシステムにアクセスを行うためのプロトコルを用いてアクセス可能な分散ファイルシステムの分散してファイルを格納している分散ファイルシステムサーバにおいて、集中型のファイルシステムにアクセスを行うためのプロトコルを分散ファイルをアクセス可能なプロトコルに変換して分散ファイルにアクセスするゲートウェイ部を備えて構成されることにより達成される。

【0011】

さらに、前記目的は、集中型のファイルシステムにアクセスを行うためのプロトコルを用いて、分散ファイルシステムにアクセスを行う分散ファイルシステムへのアクセス方法において、集中型のファイルシステムにアクセスを行うためのプロトコルを分散ファイルをアクセス可能なプロトコルに変換して分散ファイルにアクセスすることにより達成される。

【0012】

【発明の実施の形態】

以下、本発明による分散ファイルシステム（DFS）及びDFSサーバの実施形態を図面により詳細に説明する。

【0013】

図1は本発明の一実施形態による分散ファイルシステムの構成例を示すブロック図である。図1において、1はネットワーク、2はDFSサーバ、3はDFSクライアント、4はNFSクライアント、5はCIFSクライアントである。

【0014】

図1に示す本発明の一実施形態による分散ファイルシステム（DFS）は、ネットワーク1と、ネットワーク1に接続された複数のDFSサーバ2と、DFS

用のプロトコルを用いてネットワーク 1 上のファイルシステムにアクセスを行う DFS クライアント 3 と、NFS や CIFS といったネットワーク 1 上の 1 つのファイルサーバにより実現されたファイルシステムにアクセスを行うための従来プロトコルでアクセスを行う従来プロトコルクライアントである NFS クライアント 4 及び CIFS クライアント 5 とにより構成される。

【0015】

図 1 に示す DFS において、ネットワーク 1 上の DFS クライアント 3 は、専用の DFS プロトコルを用いて 1 つの DFS サーバ 2 に対してアクセスを行って、ファイルに対する参照や書き込みを行う。DFS 上のファイルには、システム内で一意に識別できる番号である GUID という識別子が付けられており、参照時に DFS クライアント 3 からファイルを特定するために用いられる。ファイルに対する GUID は、ファイルの書き込み時に、DFS サーバ 2 によって一意に割り当てられ、DFS クライアント 3 に書き込みの応答として知らされる。

【0016】

詳述すれば、DFS クライアントは、ファイルに対する参照を行う場合、ネットワーク上のいずれか 1 つの DFS サーバ 2 に対し、GUID を指定してファイルを特定してアクセスする。アクセスされた DFS サーバ 2 は、自サーバ上にファイルの実体が存在しない場合、ファイルの実体が存在する他の DFS サーバ 2 に対して問い合わせを行い、データを集めてファイルの実体を自 DFS サーバ上に構築することにより、クライアントに対してアクセスを可能とさせる。

【0017】

また、DFS クライアントは、ファイルの書き込みを行う場合、1 つの DFS サーバに対して書き込みのデータを送信する。書き込みの応答として、クライアントは、サーバから GUID を受け取る。書き込んだファイルに対するアクセスは、このサーバから受け取った GUID を用いて行う。

【0018】

DFS の場合、同一のファイルの実体が複数の DFS サーバ上に存在することがある。このような状況で、クライアントが一度書き込まれたデータの内容を変更する更新書き込みを行った場合、各 DFS サーバでファイルのデータの一貫性

を保証するのが難しい。このため、DFSの中には、GUIDで指定されるファイルへの書き込みを一度だけ行い、更新書き込みを存在させず、一度書き込んだ後は参照のみを可能とするようにしたものがある。このような特性のファイルシステムを追記型ファイルシステムと呼ぶ。

【0019】

追記型ファイルシステムの場合、更新書き込みに当たるファイルの内容の変更は、新たなファイルの世代への書き込みに当たり、その書き込み、すなわち、新たな世代に対して、新たなGUIDが割り振られて行われることとなる。そして、これらの一連の世代はファイル群として管理される。

【0020】

図1に示す本発明の実施形態によるDFSは、前述したような追記型のファイルシステムであり、一度書き込みを行い、GUIDを割り当てられたファイルに対して内容の更新を行うことができないものとする。その代わりに、図1に示すDFSは、ファイルには世代が存在し、ファイルに対する更新書き込みに当たる処理が、新しい世代のファイルを作成することに相当する。各世代には、別のGUIDが割り当てられるが、世代の異なるファイルの集まりはファイル群として管理される。このファイル群に対しても一意に識別可能なファイル群識別子が割り当てられる。そして、ファイル群識別子と世代とを示す世代番号を指定することによりファイルを識別するGUIDを得ることができる。

【0021】

図2はファイル群識別子と世代番号とからGUIDを求める方法の一例を説明する図である。図2に示すように、GUIDとして、GUIDの内上位のビットをファイル群識別子61とし、残りの下位のビットを世代番号62としたものを用いる。図2に示す例は、1つの例であり、GUIDを別の方法で求めるようにしてもよい。

【0022】

従来プロトコルクライアントであるNFSクライアント4及びCIFSクライアント5は、DFSのプロトコルではない従来のネットワークファイルシステムのプロトコルを用いてファイルにアクセスを行う。通常、これらの従来プロトコルクライアントは、ファイルにアクセスを行う場合、すでに説明したように、フ

ファイルを特定するためにディレクトリ構造を用いる。

【0023】

図3は従来プロトコルクライアントが用いるディレクトリ構造について説明する図であり、図3を参照して、ディレクトリの構造について説明する。図3の中では、ディレクトリを実線で囲んだ四角で表し、ファイルを破線で囲んだ四角で表している。

【0024】

全てのディレクトリは、ルートディレクトリ41（“/”で示される）の配下に属す木構造で定義される。そして、全てのファイルは、この木構造のディレクトリのいずれかのディレクトリ（複数であってもよい）に属することになる。このような木構造のディレクトリは、1つのディレクトリ内のファイルに対して、一意に名前（ファイル名）を付けることができる形となっている。このため、ファイルを特定する場合には、ファイルの属するディレクトリを示すパス名とそのファイルのファイル名とを指定すればよいことになる。例えば、図3に示すファイル45は、/dira/dira2/dira22/file0001という形で特定することができる。従って、クライアントは、ファイルへの参照や更新といったアクセスを行う場合、そのファイルの存在するサーバに対してパス名とファイル名とを指定してファイルを特定し、その後そのファイルの参照や更新の要求を行うことになる。

【0025】

本発明の実施形態は、クライアント側に新たなソフトウェアを組み込む等の変更を不要とし、NFSやCIFSといった従来プロトコルクライアントがDFSのファイルにアクセスを行うことを可能としており、次に、そのためのDFSサーバの構成を説明する。

【0026】

図4はDFSサーバの構成を示すブロック図、図5はゲートウェイ部の構成を示すブロック図である。図4、図5において、21はDFS制御部、22はディスクドライブ、23は主メモリ、24はOS、25はゲートウェイ部、26はDFS処理部、27はディスクドライブ処理部、28はCPU、29はHDD、31は従来プロトコル処理部、32はDFSアクセス部、33はディレクトリ管理

部である。

【0027】

DFSサーバ2は、図4に示すように、サーバ全体の処理を実行するCPU28と、主メモリ23と、ファイルを格納するディスクを備えるディスクドライブ22と、主メモリ23に展開して使用されるOS、アプリケーション等を格納するHDD29とを備えて構成される。主メモリ23内には、OS24とDFS制御部21とが構成され、また、DFS制御部21は、ゲートウェイ部25と、DFS処理部26と、ディスクドライブ処理部27とにより構成される。

【0028】

DFS制御部21は、CPU28上でDFS機能を提供するプログラムが動作することによりDFSのサーバとしての機能を果たす。そして、DFS制御部21は、ゲートウェイ部25、DFS処理部26、ディスクドライブ処理部27の処理により、ネットワーク1を介してクライアント（NSFクライアント4、CIFSクライアント5等の従来プロトコルクライアント及びDFSクライアント3）や、他のDFSサーバ2からの要求に対して処理を行う。

【0029】

DFS処理部26は、DFSプロトコルによる要求を受け付け、その要求に従ってディスクドライブ処理部27や他のDFSサーバ2に対して処理要求を行い、その結果から応答を作成して要求元に返す。

【0030】

ゲートウェイ部25は、NFSやCIFSといった従来プロトコルのネットワークファイルシステムに対するのと同じ要求をNFSクライアントやCIFSクライアントから受け付け、その要求に従った処理を必要に応じて他の処理部に要求を行って処理した後、要求に対する処理の結果をNFSクライアントやCIFSクライアントクライアントに応答として返す。

【0031】

前述により、DFSサーバ2は、従来プロトコルクライアントからの要求に対して、従来プロトコルのファイルサーバと同様にアクセスを受け、同様に、応答することが可能となる。

【0032】

ゲートウェイ部25は、全てのDFSサーバ2上に存在する必要はなく、従来プロトコルのクライアントからのアクセスポイントとなりうるDFSサーバ2上にのみ存在すればよい。すなわち、ゲートウェイ部25は、従来プロトコルのクライアントからアクセスが行われるDFSサーバにのみ必要となる。このため、通常のDFSクライアント3からアクセスを行う場合、ゲートウェイ部25は必要ではない。

【0033】

ゲートウェイ部25は、図5に示すように、従来プロトコル処理部31と、DFSアクセス部32と、ディレクトリ管理部33とにより構成される。

【0034】

従来プロトコル処理部31は、NFSクライアント、CIFSクライアントからの従来プロトコルを受け付け、従来プロトコルに合った応答を返す処理を行う処理部である。この従来プロトコル処理部31は、受け付けるプロトコル毎に存在し、対応するプロトコルの内容を判断し、要求に応じて他の処理部に処理の要求を行う。

【0035】

DFSアクセス部32は、従来プロトコル処理部31からの要求を受け付け、DFS制御部21内のDFS処理部26との橋渡しを行う処理部である。すなわち、DFSアクセス部32は、従来プロトコル処理部31からの要求に応じて、DFSに対する要求を作成し、DFS処理部26を介してDFSのファイルにアクセスを行うための処理部である。

【0036】

ディレクトリ管理部33は、従来プロトコルのファイルシステムが持っているディレクトリ構造上のファイルとDFSでのファイル識別子であるGUIDとの対応を管理するための処理部である。従来プロトコルによるファイルの指定は、そのファイルが存在するディレクトリを示すパス名とディレクトリ内で一意のファイル名とにより行われる。これに対し、DFSによるファイルの指定は、GUIDというシステム内で一意である識別子によって行われる。DFSは、従来プロトコル

でのディレクトリに当たる構造を持っている場合、その構造を使用することにより、従来ファイルシステムのファイルとGUIDとの対応を取ることが可能である。しかし、DFSの中には、ディレクトリに当たる構造を有しないシステムも存在する。すなわち、このディレクトリに当たる構造を有しないDFSは、GUIDのみで全システム内でのファイルを一意に認識することができるため、ファイルのパスに当たる構造を必要としない。

【0037】

このようなディレクトリ構造を持たないDFSは、従来プロトコルでアクセスを行うことを可能とするために、ディレクトリ構造のパスで指定されるファイルとGUIDとの対応を取るための仕組みが必要である。これを行うディレクトリ管理部33は、ディレクトリ構造のパスで指定されるファイルとGUIDとの対応を取るための仕組みを提供するものである。

【0038】

ディレクトリ管理部33は、従来プロトコルからのディレクトリ操作やディレクトリ情報の読み出し要求に対する処理を行う。このため、ディレクトリ管理部33は、従来プロトコルのファイルシステムが持っているディレクトリ管理情報と同等のディレクトリ管理情報を持つ。

【0039】

図6はディレクトリ管理情報50の構造を説明する図である。ディレクトリ管理情報50は、複数のエントリ51により構成されている。そして、エントリ51は、図3により説明したディレクトリまたはファイル毎に存在する。

【0040】

各エントリ51は、そのディレクトリまたはファイルが属するディレクトリである親ディレクトリのエントリへのポインタであるp_parent52と、そのエントリ51がディレクトリの場合、そのディレクトリに属するディレクトリまたはファイルの1つへのポインタであるp_subdir53と、同一のディレクトリのエントリ同士を指すポインタであるp_child54と、ディレクトリまたはファイルの名前を格納するf_name55とを有する。また、エントリ51がファイルの場合、DFS上でそのファイルを識別するファイル群識別子がf_id56に格納される。こ

のディレクトリ管理情報 50 の各エントリ 51 は、p_parent 52、p_subdir 53、p_child 54 の 3 種類のポインタによりディレクトリ構造に対応した構造に作成されている。そのため、パスに指定されたディレクトリのエントリについてポインタを順にたどっていくことにより目的のファイルのエントリ 51 にたどり着くことが可能である。そして、そのエントリ 51 には、ファイル群識別子が格納されているため、従来プロトコルのファイルシステムで指定するパス名とファイルから、D F S のファイル群識別子を求めることが可能である。

【0041】

すなわち、いま、図 6 の最上段に示すエントリが、図 3 に示すルートエントリであるとする、図 6 の中段に示すエントリは、図 3 に示すディレクトリ dir_a、dir_b、file1 等に相当し、最上段に示すエントリの p_subdir 53 がディレクトリ dir_a に相当する中段右側のエントリを指し示し、この中段右側のエントリの p_child 54 がディレクトリ dir_b に相当する中段右側のエントリを指し示す。同様に、ディレクトリ dir_a に相当する中段右側のエントリの p_subdir 53 がディレクトリ dir_a1 に相当する下段のエントリを指すことになる。

【0042】

従って、従来クライアントからのパス名とそのファイルのファイル名とが、図 3 で説明したように、例えば、/dir_a/dir_a2/dir_a22/file0001 と指定されたとき、ディレクトリに従って、順にエントリ 51 を辿っていくことにより、ディレクトリまたはファイルの名前を格納する f_name 55 に指定されたファイル名を持つエントリを探ことができ、これにより、ファイル群識別子 f_id 56 を得ることができる。

【0043】

前述したようなディレクトリ管理情報 50 へのファイルの登録は、次のような場合に行われる。1 つ目は、従来プロトコルでファイルを作成する場合である。この場合、パス名とファイル名とが指定されてファイルが作成されるので、従来プロトコル処理部 31 は、それに合わせてエントリを作成し、ディレクトリ管理情報 50 のパス名に当たる位置にエントリを追加すればよい。2 つ目は、D F S プロトコルでファイルを作成する場合である。この場合、そのままではディレク

トリ管理情報 50 が作成されないため、従来プロトコルからアクセスを行うことができない。そのため、DFS で作成したファイルに従来プロトコルからアクセスを行いたい場合、DFS 処理部 26 は、ゲートウェイ部 25 に対してディレクトリへの登録要求を行う必要がある。この登録要求は、DFS 処理部 26 からパス名、ファイル名及び登録するファイルのファイル群識別子を指定してゲートウェイ部 25 に行われる。

【0044】

また、前述のディレクトリ管理情報 50 は、システム内の各 DFS サーバ 2 に存在するゲートウェイ部同士で通信する等により共有することができる。これにより、各アクセスポイントから同一のファイルシステムとしてアクセスを行うことが可能となる。または、各 DFS サーバ 2 のそれぞれのゲートウェイ部 25 が別個のディレクトリ管理情報 50 を持つようにすることができ、これにより、それぞれのアクセスポイントで異なるファイルシステムとして見せることも可能である。

【0045】

図 7 は従来プロトコルを使用して DFS のファイルにアクセスしてファイルを参照するときの処理動作を説明するフローチャートであり、次に、これについて説明する。

【0046】

(1) 従来プロトコルで、あるディレクトリ内のファイルの参照を行う場合、従来プロトコルクライアント 4 または 5 は、ファイルのディレクトリ位置を示すパスとファイル名とを指定してファイルを特定し、そのファイルに対する参照を行うためのアクセスを行う。

【0047】

(2) 従来プロトコル処理部 31 は、前述のようなファイルを特定した参照要求を受けた場合、ディレクトリ管理部 33 に問い合わせを行い、ファイル群識別子を得ることになる。ディレクトリ管理部 33 は、従来のファイルシステムでディレクトリを 1 階層ずつたどると同様に、ディレクトリ管理部 33 の持つディレクトリ管理情報 50 のエントリについて順にポインタをたどって指定のファイル

のエントリにたどり着く。そして、そのエントリから指定されたファイル名のファイル群識別子を得て、従来プロトコル処理部 31 に返す（ステップ 81）。

【0048】

(3) 従来プロトコル処理部 31 は、GUIDを得るためにファイル群識別子の最新世代番号を DFS アクセス部 32 に対して問い合わせる。DFS アクセス部 32 は、DFS のプロトコルを使用することにより、ファイル群識別子を管理する他の DFS サーバ 2 に問い合わせる等の方法により最新の世代番号を求め、従来プロトコル処理部 31 に返す（ステップ 82）。

【0049】

(4) 従来プロトコル処理部 31 は、ステップ 81、82 で得たファイル群識別子と世代番号とから GUID を求める（ステップ 83）。

【0050】

(5) 従来プロトコル処理部 31 は、ステップ 83 で求めた GUID を使用して DFS アクセス部 32 に対して GUID の読み出し要求を発行する。この要求を受けた DFS アクセス部 32 は、DFS プロトコルを使用して DFS に対して要求を発行し読み出し要求に対するデータを得る（ステップ 84）。

【0051】

(6) 従来プロトコル処理部 31 は、ステップ 84 で、DFS アクセス部 32 が読み出したデータを応答として、参照要求を行ったクライアントに返す（ステップ 85）。

【0052】

図 8 は従来プロトコルを使用して DFS のファイルにアクセスしてファイルを更新するときの処理動作を説明するフローチャートであり、次に、これについて説明する。

【0053】

(1) 従来プロトコルで、あるディレクトリ内のファイルの更新を行う場合、従来プロトコルクライアント 4 または 5 は、ファイル参照時と同様に、ファイルのディレクトリ位置を示すパスとファイル名とを指定してファイルを特定し、そのファイルに対する更新を行うためのアクセスを行う。

【0054】

(2) 従来プロトコル処理部31は、前述のようなファイルを特定して更新要求を受けた場合、受け取った要求が分割されて行われる要求の2つ目以降の要求であるか否か（これについての詳細は後述する）、すなわち、すでに更新中のデータに対する更新か否かを判定する（ステップ91）。

【0055】

(3) ステップ91の判定で、すでに更新中のデータに対する更新ではなかった場合、従来プロトコル処理部31は、ディレクトリ管理部33に問い合わせを行い、ファイル群識別子を得ることになる。そのため、従来プロトコル処理部31は、参照処理の場合と同様に、ディレクトリ管理部33にファイル群識別子を問い合わせる。更新書込みの場合、既に存在するファイルに対する書込みであるため、ディレクトリ管理テーブルにエントリが存在する。参照のときと同様に、このエントリを求めることで、ファイル群識別子を得ることができる。また、ファイルの新規作成の場合、新規に作成されるファイルに対するエントリは存在しない。この場合、ファイルの作成要求時に、ディレクトリ管理情報50にエントリを作成して登録すると共に、DFS処理部26に対してもファイル作成要求を発行し、DFS処理部26からファイル群識別子を得る処理を行う必要がある。その後は、新しく作成したファイルに対して、更新時と同様に後述するステップ95からの処理を行い、データの書込みを行えばよい（ステップ92）。

【0056】

(4) ファイル群識別子を得た後、ファイルの更新内容を書き込むために、ファイル識別子の指すファイル群に新しい世代を作成する必要がある。このために、従来プロトコル処理部31は、ステップ93で、DFSアクセス部32に対して新世代作成の要求発行を依頼する（ステップ93）。

【0057】

(5) DFSアクセス部32は、DFS処理部26に対して新世代を登録し、得た世代番号を応答として従来プロトコル処理部31に返す。従来プロトコル処理部31は、この応答の世代番号とファイル群識別子からGUIDを得ることができる（ステップ94）。

【0058】

(6) 従来プロトコルは、ファイルへの一連の更新でも、1度の処理要求で行われることは限らない。例えば、NFSでは設定によりサイズの変更が可能ではあるが、通常、更新要求は、8KB単位という決まった大きさ毎のファイルの更新処理として行われ、8KBを越える更新を行う場合、複数の更新要求に分割されて行われる。例えば、24KBの更新を行いたいような場合、3つの8KBの更新に分割されて行われることとなる。このような分割されて行われる更新要求毎に新たな世代番号が作成されるのを防ぐため、一旦ファイルの更新を開始すると、その更新が終了するまでファイルが更新中であることを記憶しておく必要がある。このため、従来プロトコル処理部31は、更新を開始したファイルを後述するファイル監視テーブルに登録する（ステップ95）。

【0059】

(7) ステップ91の判定で、すでに更新中のデータに対する更新であった場合、従来プロトコル処理部31は、後述するファイル監視テーブルからGUIDを得る（ステップ96）。

【0060】

(8) 従来プロトコル処理部31は、ステップ95の処理で、ファイル監視テーブルへの登録の後、または、ステップ96の処理で、ファイル監視テーブルからGUIDを得た後、ステップ97で、GUIDを指定して更新要求のデータの書き込みをDFSアクセス部32に依頼する。DFSアクセス部32は、DFS処理部26に対して書き込みの要求を行う（ステップ97）。

【0061】

(9) 書き込み終了後、従来プロトコル処理部31は、DFS処理部26からの応答をDFSアクセス部32を介して受け取り、従来プロトコルクライアントに対して更新要求の完了の応答を返す（ステップ98）。

【0062】

図9はファイル監視テーブルの構造を説明する図である。このテーブル70の各エントリは、更新中のファイルに対応する。エントリの内容は、ファイルを識別するための情報72と、書き込み中のファイルの世代を表す世代番号73から

なる。ファイルを識別するための情報 72 として、この図 9 ではディレクトリ情報のエントリへのポインタとして p_dentry を用いている。

【0063】

受け取った要求が分割されて行われる要求の 2 つ目以降の要求であるか否かは、ファイル監視テーブル 70 に登録されている情報を調べることにより知ることができ、従来プロトコル処理部 31 は、2 つ目以降の要求である場合、新たに世代を作成しないようにする。このために、図 8 により説明した処理において、従来プロトコル処理部 31 は、更新要求を受け取った場合、ステップ 91 の処理で、ファイル監視テーブル 70 を調べ、既に更新中のファイルに対する更新要求であるか否かをチェックしている。そして、更新中でない場合、前述したように、ステップ 92 に進み、新世代を作成してデータの書き込みを行うことになる。また、既に更新中であった場合、ステップ 96 で、ファイル監視テーブル 70 から GUID を得て、その GUID に対してステップ 97 で、データの書き込み要求を行う。

【0064】

ゲートウェイ部 25 は、前述したようにしてファイルの更新要求を処理することが可能であるが、作成した新しい世代の登録を完了させるために、この世代への更新の終了を DFS 処理部 26 に通知する必要がある。このために、ゲートウェイ部 25 は、従来プロトコルクライアントから従来プロトコルで送られてくる一連の更新要求の終了を監視する必要がある。

【0065】

この一連の更新要求の終了を示すトリガは、それぞれのプロトコルにより異なる。例えば、CIFS のように、ファイルに対して操作を行うときにオープンし、終了するときにクローズを行うような、ファイル操作中の状態を持つプロトコルである場合、操作終了のクローズ要求がこの更新要求の終了を示すトリガに当たる。これに対し、NFS は、ファイルのオープンやクローズに当たる操作がなく、ファイルが操作中であるか否かを示す状態を持たない。このようなプロトコルの場合、別のトリガで操作の終了を判断する必要がある。

【0066】

この場合のトリガとしては複数の場合が考えられる。1 つは、更新要求が到着

する時間間隔である。従来プロトコルクライアントは、ゲートウェイ部 25 に対して一連の更新を分割して発行するが、通常、分割された 1 つの更新要求が終了するとすぐに次の更新要求を発行する。このため、ゲートウェイ部 25 は、更新要求が到着する時間間隔を監視し、ある一定時間以上次の更新要求が到着しない場合に、一連の更新要求が終了したとみなすことができる。また、別のトリガとしてコミットの要求が考えられる。例えば、NFS のバージョン 3 以降のものは、更新を行ったファイルの内容をディスクに反映させる方法としてコミットというコマンドが用意されている。コミットの発行契機は、特に規定されているわけではないが、通常、あるまとまった意味のある更新が終了した場合に発行される。このため、ゲートウェイ部 25 は、このコミットの要求が到着した場合に一連の更新要求が終了したとみなすことができる。但し、コミットは、更新の終了時に必ず発行されるとは限らないので、このコミットのみをトリガとすると更新の終了を認識することができない場合があるので、先に説明した時間監視と組み合わせて用いる必要がある。

【0067】

前述のようなトリガを監視して新しい世代作成の終了を判断する方法は、本来の一連の更新の終了以前にトリガだと認識し世代作成を終了してしまう可能性がある。例えば、一定時間をトリガとしている方法は、ネットワーク 1 の状態等の理由により次の更新要求の到着が遅れることにより、本来引き続いて行われている更新を終了と判断してしまう場合がある。しかし、このような場合でも、余計に世代が追加されてしまうだけで、ファイルの更新内容が失われなければ特に問題は無い。一度途切れたと判断して世代作成を終了してしまった後、更新要求を受け取った場合、また新たに世代を作成してその世代に書き込みを行えばよい。

【0068】

図 10 は世代作成終了時のゲートウェイ部 25 での処理動作を説明するフローチャートであり、次に、これについて説明する。

【0069】

ゲートウェイ部 25 は、ステップ 101 で、前述したような更新終了のトリガ発生を待ち、トリガ発生時に処理を開始する。更新終了のトリガ発生があると、

ステップ102で、従来プロトコル処理部31は、ファイル監視テーブル70を参照し、作成中の世代のGUIDを得る。ステップ103で、このGUIDに対して、DFSアクセス部32は、DFS処理部26に更新終了の要求を発行する。DFS処理部26での更新終了の処理が完了後、ステップ104で、従来プロトコル処理部31は、ファイル監視テーブル70から当該エントリを削除し、ファイルに対する更新中の状態を解除する。

【0070】

前述のようにして、ゲートウェイ部25は、従来プロトコルを用いるクライアント4、5からのDFSのファイルに対するアクセスが可能である。

【0071】

前述した本発明の実施形態による各処理は、処理プログラムとして構成することができ、この処理プログラムは、HD、DAT、FD、MO、DVD-ROM、CD-ROM等の記録媒体に格納して提供することができる。

【0072】

前述で説明した実施形態は、DFSとして追記型のファイルシステムを前提として説明したが、本発明は、追記型でないDFSに対しても適用することが可能である。この場合、ファイルの世代という考えはなく、ファイルが更新されてもGUIDの値は変化しない。そのため、ディレクトリ管理情報50のf_id56にファイル群識別子の代わりに直接ファイルのGUIDを持つようにすればよい。そして、図7におけるファイルの参照時のフローでは、ステップ81でGUIDを得ることができ、ステップ82とステップ83とは不要となる。同様に、図8に示すファイルの更新時のフローでは、ステップ92でGUIDを得ることになり、世代作成のステップ93とGUIDを得るステップ94とは不要となる。

【0073】

また、前述した本発明の実施形態は、ゲートウェイ部25をDFSサーバ2内に内蔵させるとして説明したが、本発明は、ゲートウェイ部25をDFSサーバ2内に実装しなくてもよい。例えば、他の実装位置として、ネットワーク1上にゲートウェイ部25の処理を行うゲートウェイサーバを持つ場合や、従来プロトコルクライアント4や5内に組み込むことが考えられる。

【0074】

図11は本発明の他の実施形態による分散ファイルシステムの構成例を示すブロック図である。図11において、110はゲートウェイサーバであり、他の符号は図1の場合と同一である。

【0075】

図11に示す例は、前述で説明したゲートウェイ部25をネットワーク1上に設けたゲートウェイサーバ110として備えた例である。この場合、従来プロトコルクライアント4や5は、ゲートウェイサーバ110に対してアクセスを行うことになる。そして、ゲートウェイサーバ110は、DFSサーバ2に対して、DFSプロトコルを用いてアクセスを行う。ゲートウェイサーバ110は、ゲートウェイ部25と同等の処理を行う。

【0076】

前述したゲートウェイ部25のDFSアクセス部32は、DFS処理部26と連携して処理を行っていたが、図11に示す例の場合、DFSサーバ2と別のサーバ上にゲートウェイ部の機能が実装されるため、前述した連携した処理をおこなうことが不可能となる。その代わりに、図11に示す例のシステムは、ゲートウェイサーバ110内に設けられるDFSアクセス部32がDFSクライアントとなり、DFSプロトコルを使用してDFSサーバ2に要求を行うことにより同等の処理を行うことが可能である。

【0077】

また、従来プロトコルクライアントに組み込む方法としては、例えば、NFSのプロトコル処理部にこのゲートウェイ部25の機能を組み込むことが考えられる。この場合、上位のNFSを使用するプログラムは、NFSのプロトコルを用いて処理を行って、ゲートウェイ部25の機能がNFSのプロトコルをDFSプロトコルに変換し、ネットワーク1上ではDFSのプロトコルを用いてDFSサーバ2にアクセスを行うこととなる。

【0078】

前述した本発明の実施形態によれば、NFSやCIFS等の従来プロトコルのファイルシステムに存在するディレクトリ構造をエミュレートし、ディレクトリ

構造上でファイルの存在する位置を示すパス名とファイルの名前であるファイル名とによりアクセスを行う方式を、D F Sでのファイル識別子であるGUIDによるアクセスに変換することにより、従来プロトコルを使用してD F S上のファイルに対するアクセスを行うことができる。

【 0 0 7 9 】

また、本発明の実施形態によれば、D F Sが追記型のファイルシステムである場合、従来プロトコルの更新処理を新しい世代の作成処理に変換することにより、参照時には、世代管理されているファイル群から最新の世代のファイルを求め、そのファイルのデータにアクセスを行うことが可能となる。

【 0 0 8 0 】

【発明の効果】

以上説明したように本発明によれば、クライアント側の変更を行うことなく、N F SやC I F S等の従来プロトコルを用いてD F S上のファイルにアクセスを行うことができる。

【図面の簡単な説明】

【図 1】

本発明の一実施形態による分散ファイルシステムの構成例を示すブロック図である。

【図 2】

ファイル群識別子と世代番号とからGUIDを求める方法の一例を説明する図である。

【図 3】

従来プロトコルクライアントが用いるディレクトリ構造について説明する図である。

【図 4】

D F Sサーバの構成を示すブロック図である。

【図 5】

ゲートウェイ部の構成を示すブロック図である。

【図 6】

ディレクトリ管理情報 5 0 の構造を説明する図である。

【図 7】

従来プロトコルを使用して D F S のファイルにアクセスしてファイルを参照するときの処理動作を説明するフローチャートである。

【図 8】

従来プロトコルを使用して D F S のファイルにアクセスしてファイルを更新するときの処理動作を説明するフローチャートである。

【図 9】

ファイル監視テーブルの構造を説明する図である。

【図 1 0】

世代作成終了時のゲートウェイ部 2 5 での処理動作を説明するフローチャートである。

【図 1 1】

本発明の他の実施形態による分散ファイルシステムの構成例を示すブロック図である。

【符号の説明】

- 1 ネットワーク
- 2 D F S サーバ
- 3 D F S クライアント
- 4 N F S クライアント
- 5 C I F S クライアント
- 2 1 D F S 制御部
- 2 2 ディスクドライブ
- 2 3 主メモリ
- 2 4 O S
- 2 5 ゲートウェイ部
- 2 6 D F S 処理部
- 2 7 ディスクドライブ処理部
- 2 8 C P U

2 9 H D D

3 1 従来プロトコル処理部

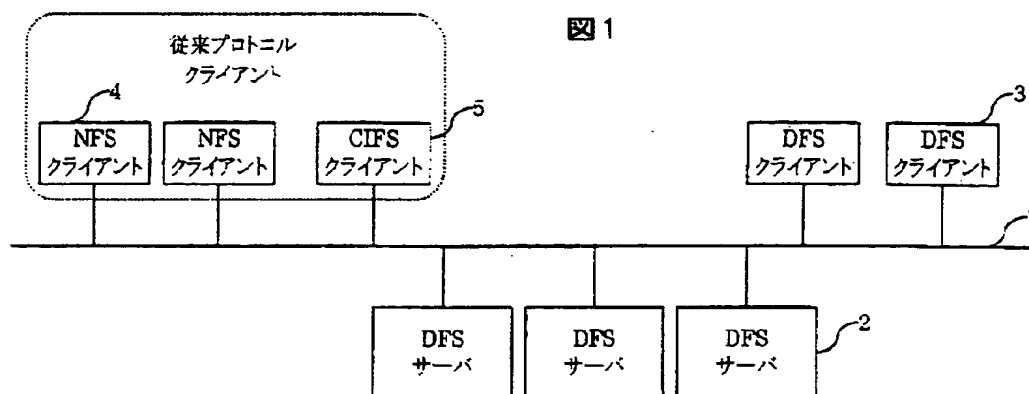
3 2 D F S アクセス部

3 3 ディレクトリ管理部

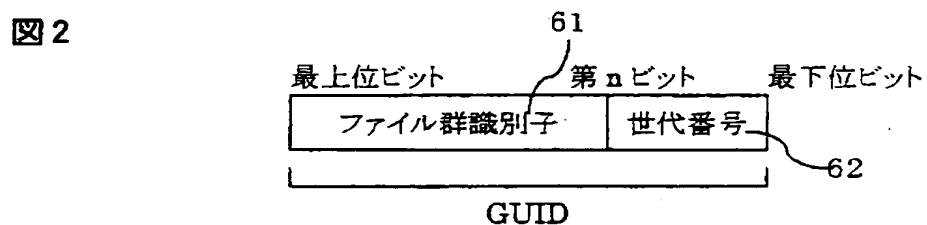
1 1 0 ゲートウェイサーバ

【書類名】 図面

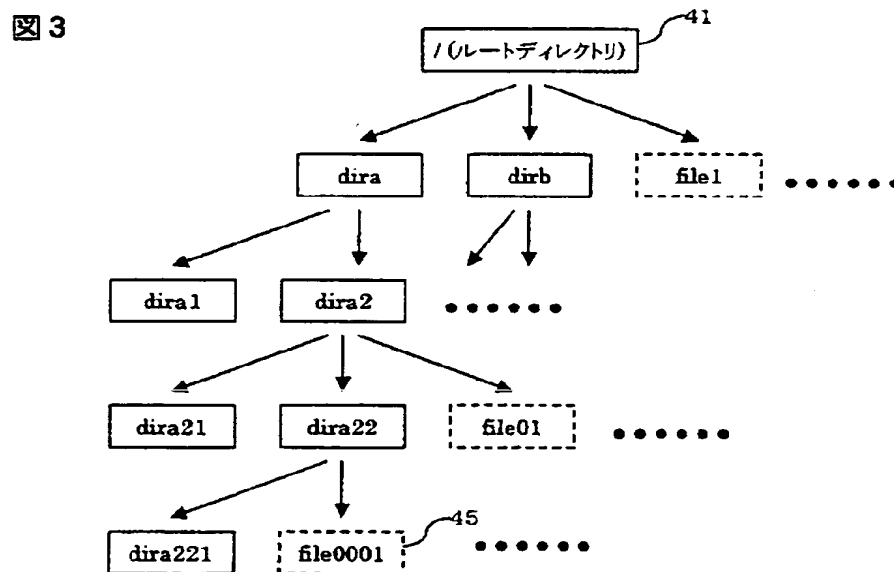
【図 1】



【図 2】

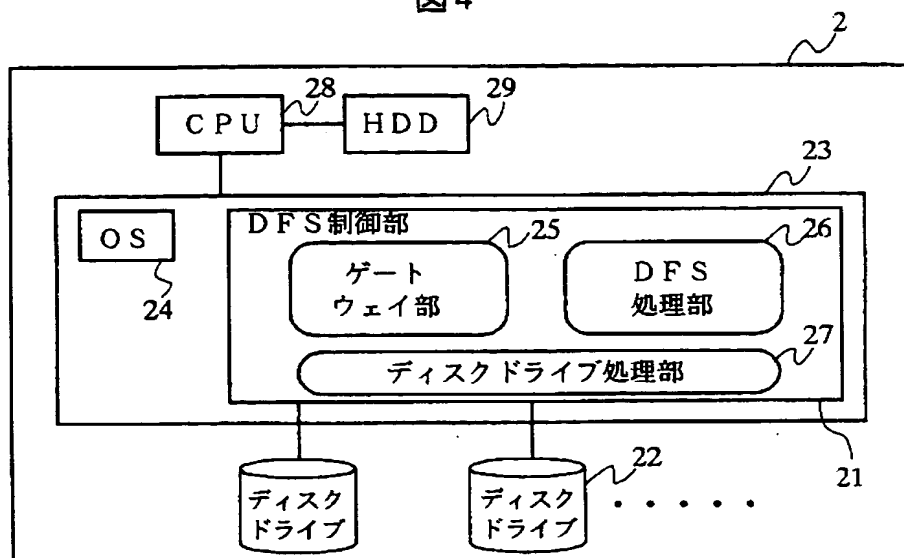


【図 3】



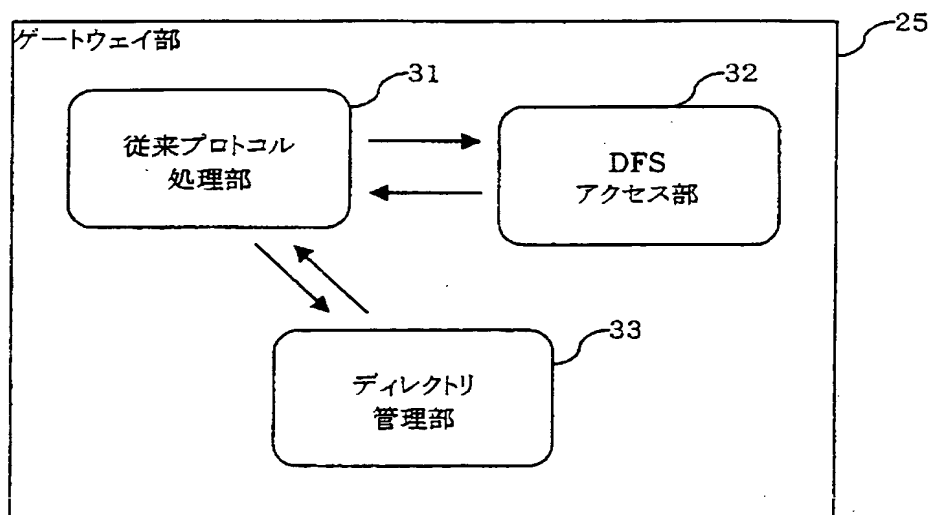
【図 4】

図 4

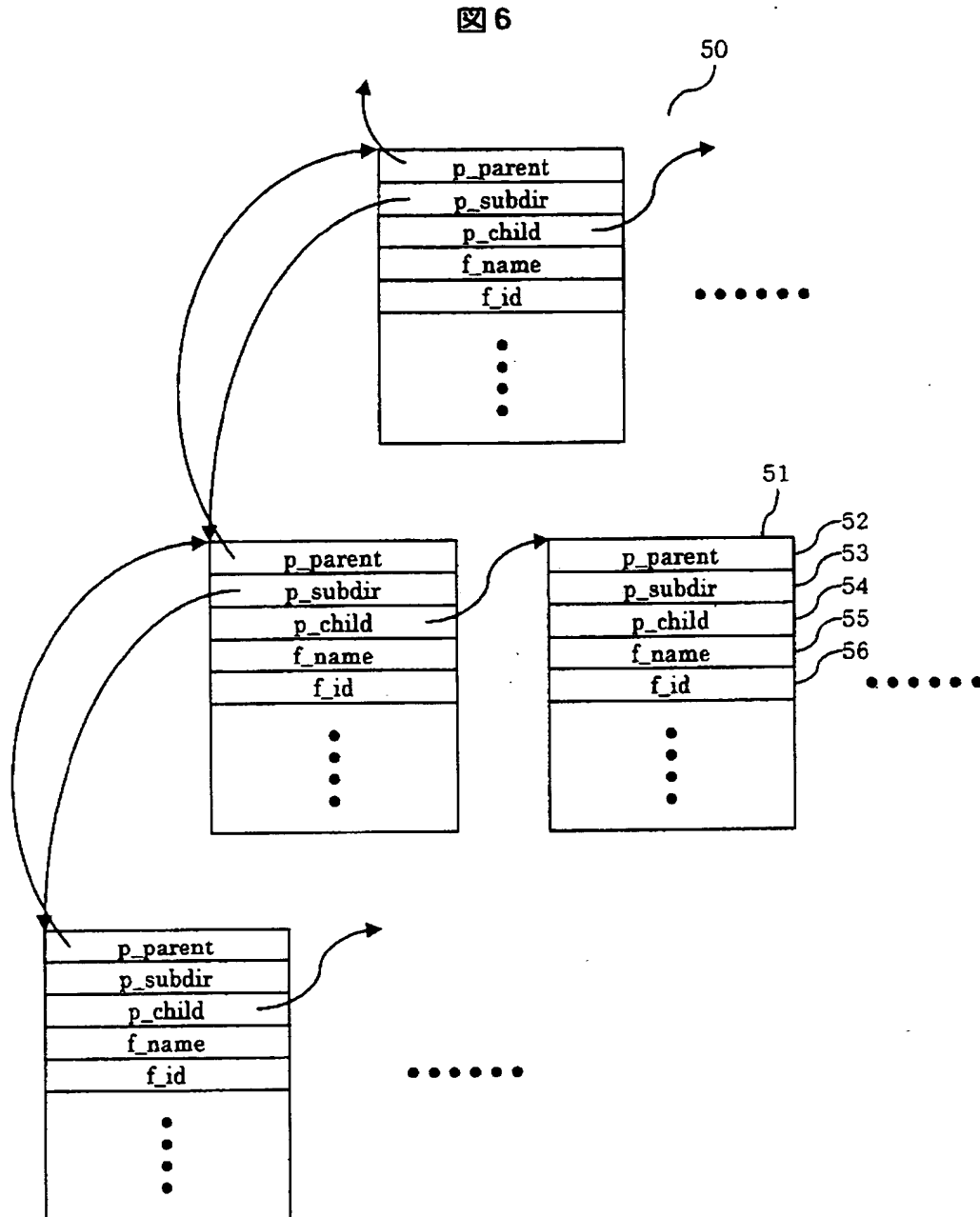


【図 5】

図 5



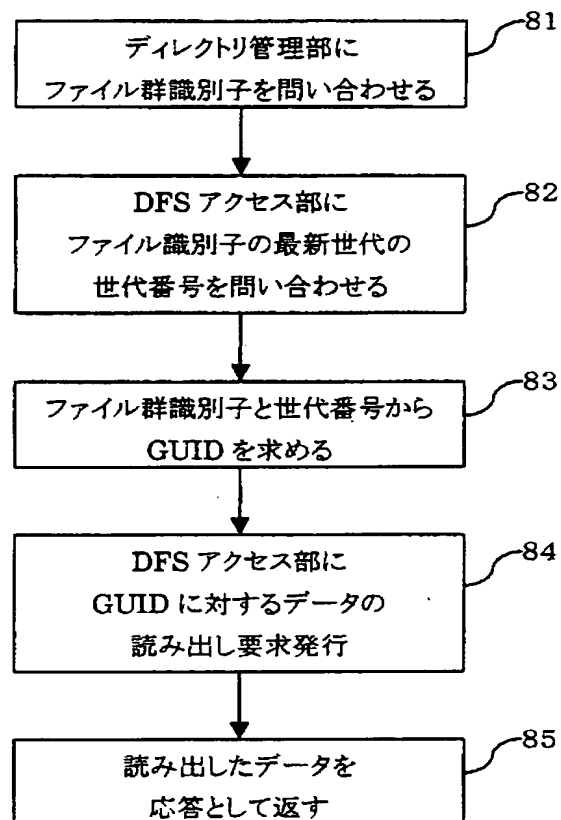
【図 6】



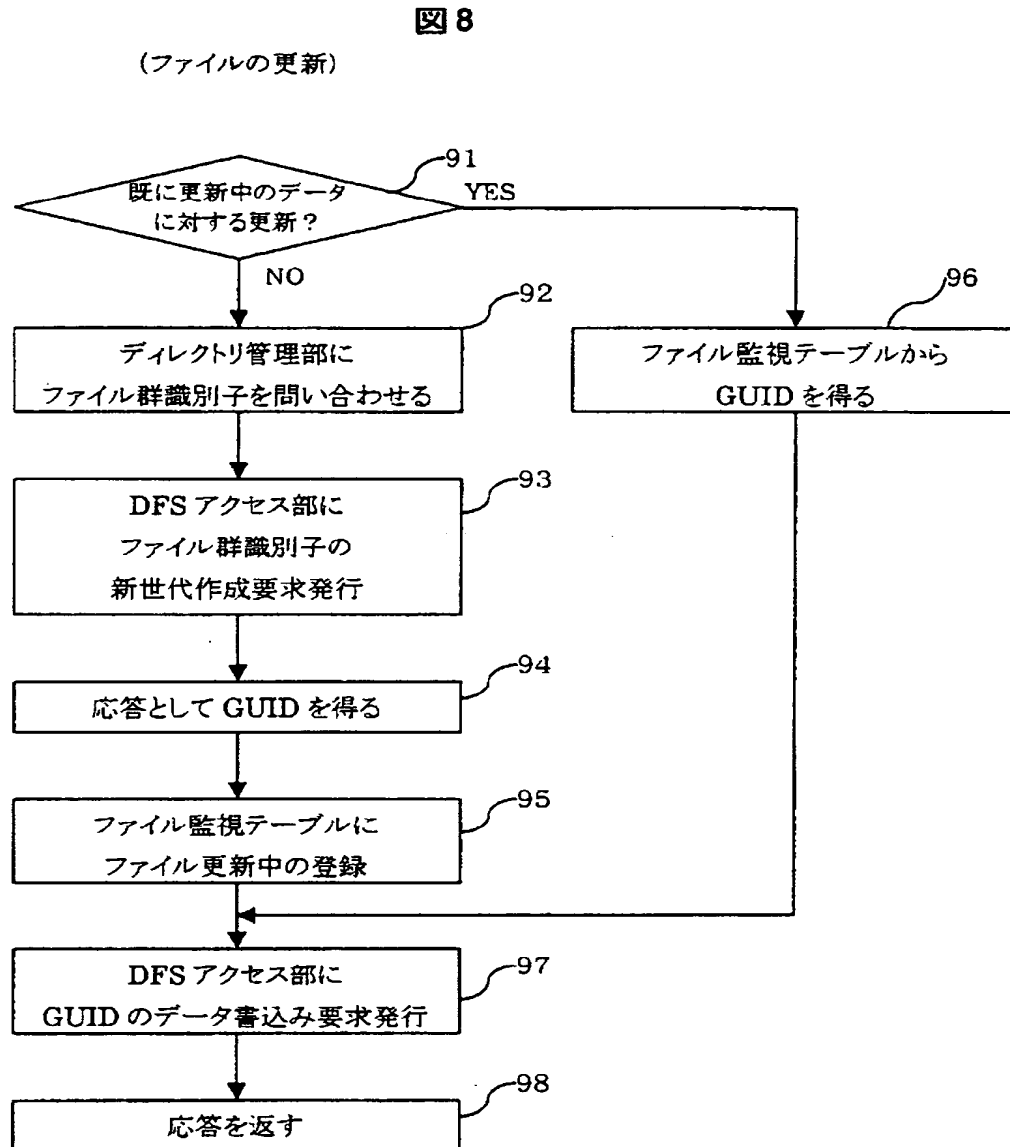
【図 7】

図 7

(ファイルの参照)



【図 8】



【図 9】

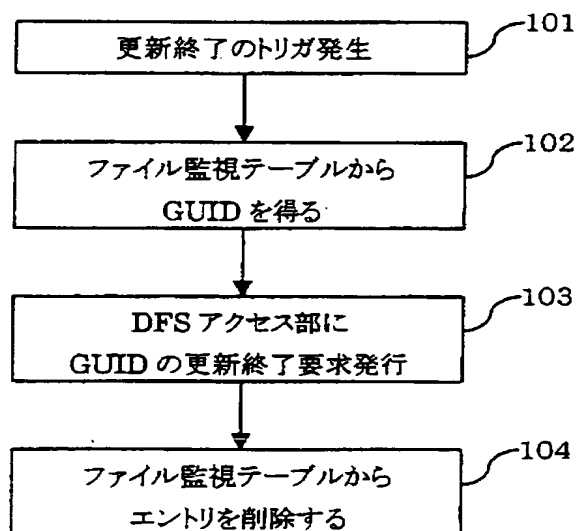
図 9

p_dentry	generation	70
p_dentry	generation	
p_dentry	generation	73
p_dentry	generation	
⋮		

【図 10】

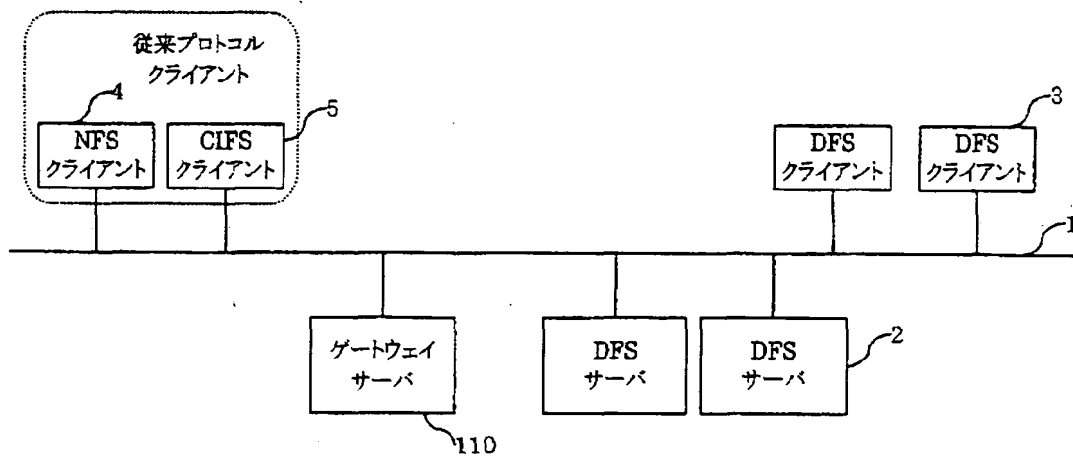
図 10

(世代作成終了処理)



【図 11】

図 11



【書類名】 要約書

【要約】

【課題】 クライアント側の変更を行うことなく、N F S や C I F S 等の従来プロトコルを用いて分散ファイルシステム（D F S）にアクセスを行うこと。

【解決手段】 D F S サーバに従来プロトコルを受け付け、それに対応する処理を D F S に対して行うゲートウェイ部 2 5 を設ける。ゲートウェイ部 2 5 は、N F S や C I F S 等のファイルシステムに存在するディレクトリ構造をエミュレートする。また、D F S が追記型ファイルシステムである場合に、更新処理を新しい世代の作成処理に変換し、参照処理を世代管理されているファイル群の最新世代へのアクセスに変換する。そして、ゲートウェイ部 2 5 は、D F S 処理部 2 6 を介してファイルにアクセスする。

【選択図】 図 4

特願 2 0 0 3 - 0 6 8 5 5 3

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 1 0 8]

1. 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台 4 丁目 6 番地

氏 名

株式会社日立製作所